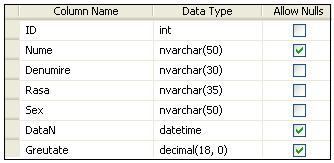
1. Se creeze o aplicaţie de tip **WindowsFormsApplication** numită **FermaVesela**.
2. Pentru a crea baza de date, click dreapta pe numele proiectului, **Add →** **New** I**tem** şi alegeţi **Service-based Database**, daţi numele **ferma.mdf** şi apăsaţi butonul **Add**. Aşteptaţi puţin până apare fereastra **Data Source Configuration Wizard**, se va crea **DataSet**-ul cu numele implicit **fermaDataSet**. Apăsaţi butonul **Finish**.
3. În tab-ul **Database Explorer** **(**apare de obicei unde este şi Toolbox-ul. Dacă nu apare, alegeţi din meniul **View**, opţiunea **Other Windows** şi apoi alegeţi **Database Explorer)**, expandaţi nodul **ferma.mdf**, daţi **click dreapta pe Tables** şi alegeţi **Add New Table** pentru a adăuga o nouă tabelă în baza de date. Completaţi structura tabelei având câmpurile următoare:



Pentru câmpul ID setaţi proprietatea Is Identity la Yes. Pentru acest câmp nu veţi completa valorile acestea se vor completa automat.

1. Salvaţi tabela cu numele **animale (**apăsând butonul **Save).** Salvați și proiectul cu numele **FermaVesela,** apăsând butonul **Save All.**
2. În fereastra **Database Explorer**, daţi **click dreapta pe** tabela **animale** şi alegeţi opţiunea **Show Table Data.** Adaugați câteva inregistrari (să aveți mai multe înregistrări cu aceeași denumire, și câteva cu aceeași rasă). De ex. :

În fereastra **Solution Explorer**, selectați baza de date **ferma.mdf** și în fereastra **Properties** modificați valoarea proprietății **Copy to Output Directory** la ***Copy if newer***.

1. Din meniul **Data**, selectaţi opţiunea **Show Data Sources**. În fereastra **Data Sources**, daţi **click dreapta pe** elementul **firmaDataSet**, apoi selectaţi **Configure DataSet with Wizard…** În dialogul **Data Source Configuration Wizard**, selectaţi **Tables**, apoi apăsaţi butonul **Finish**.

Pe forma **Form1**, adăugaţi din **Toolbox** un control de tip **MenuStrip** şi completaţi cu opţiunile:

**Operatii** (va avea un submeniu cu opțiunile: Adaugare animal, Modificarea greutatii unui animal, Stergere)

**Afișări** (va avea un submeniu cu opțiunile: Ordonate dupa denumire și rasa, Ordonate dupa data nasterii, Ordonate crescător după rasă, apoi după greutate descrescător) ***.***

**Cautari** (va avea un submeniu cu opțiunile: dupa Denumire, dupa Nume)

**Ieșire**

Pentru a trata opțiunea **Ieșire**, care va încheia aplicația, dați dublu click pe aceasta și scrieți codul:

Application.Exit();

Adăugați proiectului o nouă formă, numită **Adaugare**. Pentru aceasta, în fereastra **Solution Explorer**, dați click dreapta pe numele proiectului, alegeți opțiunea **Add**, apoi **Windows Form …** și dați numele **Adaugare**.

În fereastra **Data Source**, selectaţi tabela **animale**, deschideţi lista ascunsă din dreptul ei şi selectaţi opţiunea **Details**. Selectaţi apoi tabela şi trageţi-o pur şi simplu pe forma **Adaugare**. Vor apărea automat controale de tip TextBox şi DateTimePicker (legate la tabela animale) în care vor putea fi introduse informaţii noi, precum și alte controale care ne leagă de baza de date și care apar în try-ul formei.

Deoarece în baza de date câmpul ID se completează cu date automat ( are setată proprietatea Is Identity la Yes) , vom șterge label-ul și textBox-ul atașate. De asemenea vom șterge de pe formă și animaleBindingNavigator (cu ajutorul acestuia puteam parcurge sau face diferite operații cu înregistrările din tabelă, dar noi nu avem nevoie acum de acesta. Adăugarea informațiilor o vom face prin accesarea unui buton pe care îl vom adăuga ulterior).

Ștergeți controlul **sexLabel** și **sexTextBox** în care ar trebui să întroduceți sexul animalului. În loc de acestea, adăugați pe formă un control de tip **groupBox**. Pentru acesta în fereastra **Properties** setați proprietatea **Text** la ***Sex:*** . Adăgați în **groupBox1**, două controale de tip **radioButton** pentru care setați proprietatea **Name** la ***fRadioButton***, respectiv **mRadioButton**, iar proprietatea **Text** la ***Feminin***, respectiv la ***Masculin***.

În metoda Load a formei **Adăugare** (dați dublu click pe formă unde nu avem un alt control, pentru a scrie cod în ea) adăugați codul următor pentru a inițializa controalele de tip **TextBox** cu șirul vid și pentru a fi selectat implicit sexul ca fiind feminin prin intermediul controlului **fRadioButton**.

numeTextBox.Text = "";

denumireTextBox.Text = "";

rasaTextBox.Text = "";

greutateTextBox.Text = "";

fRadioButton.Checked = true;

Pentru a adăuga o nouă înregistrare în tabela **animale**, a căror informaţii sunt date în fereastra **Adăugare** , se va realiza o ***interogare***. Pentru aceasta procedaţi în felul următor:

* În fereastra **Data Sources**, daţi **click dreapta pe** tabela **animale** şi alegeţi opţiunea **Edit DataSet with Designer.**
* Daţi **click dreapta pe animaleTableAdapter** şi alegeţi opţiunea **Add → Query.** Va apărea fereastra TableAdapter Query Configuration Wizard, în care este selectată implicit opţiunea **Use SQL statements**. Apăsaţi butonul **Next**.
* Va trebui să alegeţi ce comandă SQL vreţi să folosiţi, alegeți **INSERT**. Apăsaţi butonul **Next**.
* Se va afişa implicit o comandă **INSERT** care adaugă o nouă înregistrare în tabela **animale**. Pentru a o modifica sau testa această comandă apăsaţi butonul **Query Builder…** Comanda folosită este dată mai jos (observați că nu se completeayă acum cțmpul ID, deoarece acesta este de tip Identity și va fi completat automat):

**INSERT INTO animale**

**(Nume, Denumire, Rasa, Sex, DataN, Greutate)**

**VALUES (@Nume,@Denumire,@Rasa,@Sex,@DataN,@Greutate)**

Apăsaţi butonul **OK,** apăsaţi butonul **Next**. Observaţi că automat s-a generat metoda **InsertQuery** (acum lăsaţi pentru această metodă numele generat implicit, în viitor dacă doriţi puteţi schimba numele cu altul ) .

* Apăsaţi butonul **Next** şi apoi **Finish** pentru a termina interogarea.

Reveniți pe forma **Adăugare** și adăugați din **Toolbox** un control de tip **Button**. Modificați proprietatea **Name** a acestuia la ***addButton***, iar proprietatea **Text** a acestuia la ***Adăugati*** !. Când veți da click pe acest buton se va adăuga în tabelă o nouă înregistrare cu informațiile completate în formă. Pentru a trata evenimentul Click al butonului veți scrie codul de mai jos:

string sex;

if (fRadioButton.Checked)

sex="f";

else

sex="m";

animaleTableAdapter.InsertQuery(numeTextBox.Text,denumireTextBox.Text,rasaTextBox.Text,sex,DateTime.Parse(dataNDateTimePicker.Text),decimal.Parse(greutateTextBox.Text));

Close(); //închide forma

Reveniţi în **Form1** şi pentru a trata evenimentul **Click** al opţiunii ***Adaugare animal*** veți scrie codul următor:

Adaugare ad = new Adăugare();

ad.ShowDialog(); // afiseaza Form2, ca dialog modal acest tip de

// fereastra dialog nu permite posibilitatea accesarii altor

//ferestre ale aplicatiei pana cand acest dialog nu este inchis

Pe Form1 adăugaţi din Toolbox un control de tip Button. Setaţi atât proprietatea Name cât şi proprietatea Text la Refresh. Pentru tratarea evenimentului Click al acestui buton scrieţi codul următor care va reactualiza datele despre animale:

private void Refresh\_Click(object sender, EventArgs e)

{

this.animaleTableAdapter.Fill(this.fermaDataSet.animale);

}

Pentru a modifica greutatea unui animal cu ID-ul dat sau pentru a-l şterge, parcurgeţi paşii următori:

* Veți crea două noi interogări. Veți proceda asemănător ca la Adăugare doar că nu veți alege comanda INSERT ci veți folosi pentru modificare comanda UPDATE și pentru ștergere comanda DELETE.

Pentru a modifica greutatea unui animal cu ID-ul dat cu o altă greutate dată, vom folosi comanda:

**UPDATE animale**

**SET Greutate=@gr**

**WHERE ID=@id**

iar metoda generată redenumiţi-o cu numele **UpdateGrQuery**.

Pentru a șterge un animal cu ID-ul dat, vom folosi comanda:

**DELETE animale**

**WHERE ID=@id**

iar metoda generată redenumiţi-o cu numele **DeleteIDQuery**.

* Adăugaţi pe Form1, sub controlul de tip DataGridView, un control de tip groupBox şi setaţi pentru acesta proprietatea Name la ModifStergGroupBox. Pe el adăugați:
  + un control de tip Label pentru care setaţi proprietatea Text la: ***Daţi ID-ul:*** şi proprietatea ***Name*** la ***IDLabel.***
  + un control de tip ***TextBox*** pentru care setaţi proprietatea Name la: IDTextBox
  + un control de tip Label pentru care setaţi proprietatea Text la: ***Dati noua greutate:*** şi proprietatea ***Name*** la ***GrLabel.***
  + un control de tip ***TextBox*** pentru care setaţi proprietatea Name la: GrTextBox
  + un control de tip Button pentru care setaţi proprietatea Name la: ModifStergButton
* iniţial, când se încarcă Form1, ModifStergGoupBox nu va fi vizibil. Pentru aceasta, în metoda Load a formei scrieţi codul:

ModifStergGroupBox.Visible = false;

* Pe **Form1**, deschideţi submeniul **Operatii** din **menuStrip1** şidaţi dublu click pe opţiunea ***Modificarea greutății unui animal*** pentru a trata evenimentul **Click** şi scrieţi codul următor:

ModifStergGroupBox.Visible = true;

ModifStergGroupBox.Text = "Modificarea greutatii animalului cu

ID-ul dat !";

ModifStergButton.Text = "Modifica";

* Pe **Form1**, deschideţi submeniul **Operatii** din **menuStrip1** şidaţi dublu click pe opţiunea ***Stergere*** pentru a trata evenimentul **Click** şi scrieţi codul următor:

ModifStergGroupBox.Visible = true;

ModifStergGroupBox.Text = "Stergerea animalului cu ID-ul dat";

GrLabel.Visible = false;

GrTextBox.Visible = false;

ModifStergButton.Text = "Sterge";

* Scrieţi codul de mai jos pentru tratarea evenimentului Click a butonului **ModifStergButton** :

private void ModifStergButton\_Click(object sender, EventArgs e)

{

if (ModifStergButton.Text == "Modifica")

{

if (IDTextBox.Text == "" || GrTextBox.Text == "")

MessageBox.Show("Completati ID si noua greutate !");

else

{

animaleTableAdapter.UpdateGrQuery(decimal.Parse(GrTextBox.Text),

int.Parse(IDTextBox.Text));

ModifStergGroupBox.Visible = false;

}

}

else

{

if (IDTextBox.Text == "")

MessageBox.Show("Completati ID-ul !");

else

{

animaleTableAdapter.DeleteIDQuery(int.Parse(IDTextBox.Text));

ModifStergGroupBox.Visible = false;

}

}

}

Pentru a afisa animalele ordonate în funcţie de optiunile din submeniul Afisări, parcurgeţi paşii de mai jos:

Reveniţi pe Form1.

În fereastra **Data Source**, selectaţi tabela **animale**, deschideţi lista ascunsă din dreptul ei şi selectaţi opţiunea **DataGridView**. Selectaţi apoi tabela şi trageţi-o pur şi simplu pe **Form1.** Va apărea automat un control de tip **DataGridView** (legat la tabela animale), în care vor putea fi afișate informaţile despre animale, precum și alte controale care ne leagă de baza de date și care apar în try-ul formei. *Stergeţi* de pe formă **animaleBindingNavigator**.

Vom crea acum cele trei interogări cu ajutorul cărora vom afișa în controlul de tip **DataGridView** informațiile din tabela **animale**, ordonate după opţiunile din submeniul **Afişări**. Procedați astfel:

* În fereastra **Data Sources**, daţi **click dreapta pe** tabela **animale** şi alegeţi opţiunea **Edit DataSet with Designer.**
* Daţi **click dreapta pe animaleTableAdapter** şi alegeţi opţiunea **Add → Query.** Va apărea fereastra TableAdapter Query Configuration Wizard, în care este selectată implicit opţiunea **Use SQL statements**. Apăsaţi butonul **Next**.
* Va trebui să alegeţi ce comandă SQL vreţi să folosiţi. Va fi selectată implicit comanda care ne trebuie: **SELECT wich returns rows**. Apăsaţi butonul **Next**.
* Se va afişa implicit o comandă **SELECT** care întoarce toate înregistrările din tabela **animale**. Aceasta va trebui modificată. Pentru aceasta, apăsaţi butonul **Query Builder…** şi modificaţi comanda ca mai jos:

**SELECT ID, Nume, Denumire, Rasa, Sex, DataN, Greutate**

**FROM animale**

**ORDER BY Denumire, Rasa**

apoi apăsaţi butonul **Execute Query** pentru a verifica corectitudinea comenzii. Dacă comanda e bună, apăsaţi butonul **OK,** apăsaţi butonul **Next**. Redenumiţi metodele generate cu numele **FillByDenRasa** şi **GetDataByDenRasa.** Cu ajutorul acestora veţi face legătura dintre interogarea voastră şi controalele în care afişaţi rezultatul interogării .

* Apăsaţi butonul **Next** şi apoi **Finish** pentru a termina interogarea.

Procedaţi la fel ca mai sus pentru a crea celelalte două interogări.

A doua interogare cu ajutorul căreia obţinem datele din tabelă ordonate crescător după data naşterii va avea codul:

**SELECT ID, Nume, Denumire, Rasa, Sex, DataN, Greutate**

**FROM animale**

**ORDER BY DataN**

iar metodele generate redenumiţi-le cu numele **FillByDataN** , respective **GetDataByDataN**.

A treia interogare cu ajutorul căreia obţinem datele din tabelă ordonate crescător după rasă, apoi după greutate descrescător, va avea codul:

**SELECT ID, Nume, Denumire, Rasa, Sex, DataN, Greutate**

**FROM animale**

**ORDER BY Rasa, Greutate DESC**

iar metodele generate redenumiţi-le cu numele **FillByRasaGr** , respectiv **GetDataByRasaGr**

Pe **Form1**, deschideţi submeniul **Afişări** din m**enuStrip1** şidaţi dublu click pe opţiunea ***Ordonate dupa denumire și rasa*** pentru a trata evenimentul **Click** şi scrieţi codul următor pentru a afişa în controlul de tip **DataGridView** animalele ordonate conform opţiunii. Veţi afişa de fapt rezultatul interogării realizate anterior.

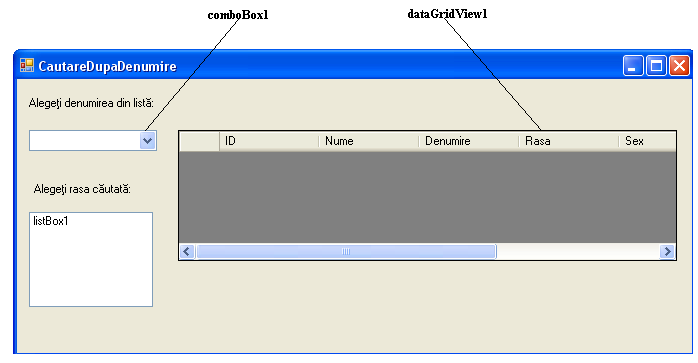
animaleTableAdapter.FillByDenRasa(fermaDataSet.animale);

În acelaşi submeniu daţi dublu click pe opţiunea ***Ordonate dupa data nasterii*** şi scrieţi codul următor:

animaleTableAdapter.FillByDataN(fermaDataSet.animale);

apoi daţi dublu click pe opţiunea ***Ordonate crescător după rasă, apoi după greutate descrescător*** şi scrieţi codul următor:

animaleTableAdapter.FillByRasaGr(fermaDataSet.animale);

Adăugaţi proiectului o nouă formă numită **CautareDupaDenumire**. Adăugati pe ea controalele din imagine:

Pentru a putea căuta în baza de date, parcurgeți următorii pași:

Pe formă, adăugați din **Toolbox** (din prima secțiune) un control de tipul **animaleTableAdapter** și un control **fermaDataSet**. Acestea se vor adăuga automat în **try**-ul formei și vor avea numele implicite: ***animaleTableAdapter1*** și ***fermaDataSet1***.

Pentru a încărca în comboBox1 toate denumirile animalelor existente în tabela animale:

* scrieți metoda de mai jos care va adăuga în comboBox1 un sir de caractere, primit ca parametru, doar dacă acesta nu mai există în listă:

**void IncarcaComboBoxFaraDuplicate(string s)**

**{**

**for (int i = 0; i < comboBox1.Items.Count; i++)**

**if (comboBox1.Items[i].ToString() == s)**

**return;**

**comboBox1.Items.Add(s);**

**}**

* în metoda **Load** a formei scrieți codul următor care încarcă comboBox-ul:

**private void CautareDupaDenumire\_Load(object sender, EventArgs e)**

**{**

**this.animaleTableAdapter1.Fill(this.fermaDataSet1.animale);**

**string s;**

**int i=0;**

**while (i < fermaDataSet1.animale.Rows.Count)**

**{**

**s = fermaDataSet1.animale.Rows[i][2].ToString();**

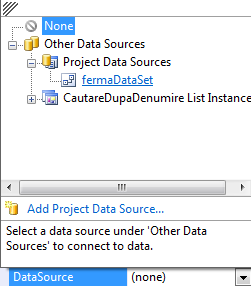
**IncarcaComboBoxFaraDuplicate(s);**

**i++;**

**}**

**}**

Tot din **ToolBox**, din secțiunea **Data**, adăugați un control de tip **BindingSource**. Acesta va avea numele implicit ***bindingSource1***.

Selectați controlul **bindingSouce1** din **try**-ul formei, iar în fereastra **Properties**, modificați proprietatea **DataSource** la **fermaDataSet**, expandând nodurile ca în imaginea alăturată, apoi alegeți pentru proprietatea **DataMember** tabela **animale**.

Pentru a încărca în **listBox1** toate rasele animalelor existente în tabela animale a căror denumire a fost selectată în comboBox1:

* scrieți metoda de mai jos care va adăuga în listBox1 un sir de caractere, primit ca parametru, doar dacă acesta nu mai există în listă:

**void IncarcaListBoxFaraDuplicate(string s)**

**{**

**for (int i = 0; i < listBox1.Items.Count; i++)**

**if (listBox1.Items[i].ToString() == s)**

**return;**

**listBox1.Items.Add(s);**

**}**

* pentru a trata evenimentul **selectedIndexChanged** al controlului **comboBox1** scrieți codul următor care afișează în DataGridView datele animalelor cu denumirea selectată în comboBox și încarcă în listBox1 doar rasele animalelor cu denumirea selectată în comboBox:

**private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)**

**{**

**bindingSource1.Filter = "Denumire='" +**

**comboBox1.SelectedItem.ToString() + "'";**

// deoarece DataGridView1 e legat de bindingSource1, se vor afișa datele filtrate

**listBox1.Items.Clear();**

**string s;**

**int i = 0;**

**while (i < fermaDataSet1.animale.Rows.Count)**

**{**

**if (fermaDataSet1.animale.Rows[i][2].ToString() == comboBox1.SelectedItem.ToString())**

**{**

**s = fermaDataSet1.animale.Rows[i][3].ToString();**

**IncarcaListBoxFaraDuplicate(s);**

**}**

**i++;**

**}**

**}**

Pentru a filtra datele în **dataGridView1**, când selectați rasa în **listBox1**, tratați evenimentul **SelectedIndexChanged** al acestui control scriind codul de mai jos:

**private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)**

**{**

**bindingSource1.Filter = "Denumire='" + comboBox1.SelectedItem.ToString() + "' AND Rasa='"+listBox1.SelectedItem.ToString()+"'";**

**}**

Reveniţi în **Form1** şi pentru a trata evenimentul **Click** al opţiunii ***Dupa denumire***, din meniul **Cautari**,veți scrie codul următor:

private void dupaDenumireToolStripMenuItem\_Click(object sender, EventArgs e)

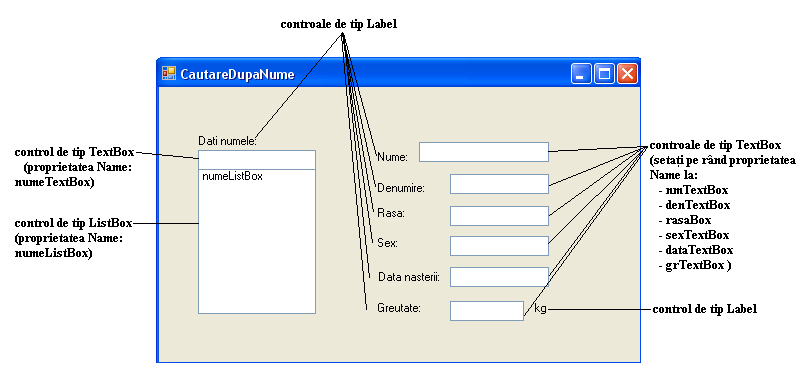
{

CautareDupaDenumire fc = new CautareDupaDenumire();

fc.ShowDialog();

}

Adăugaţi proiectului o nouă formă numită **CautareDupaNume**. Adăugati pe ea controalele din imagine:



Pentru a putea căuta în baza de date, parcurgeți următorii pași:

Pe formă, adăugați din **Toolbox** (din prima secțiune) un control de tipul **animaleTableAdapter** și un control **fermaDataSet**. Acestea se vor adăuga automat în **try**-ul formei și vor avea numele implicite: ***animaleTableAdapter1*** și ***fermaDataSet1***.

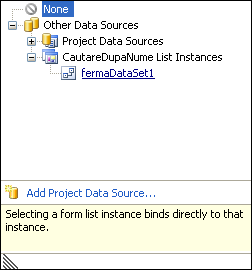
Pentru a încărca datele din tabela animale în fermaDataSet1, în metoda **Load** a formei scrieți codul următor :

private void CautareDupaNume\_Load(object sender, EventArgs e)

{

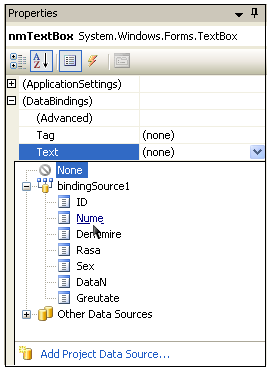
this.animaleTableAdapter1.Fill(this.fermaDataSet1.animale);

}

Tot din **ToolBox**, din secțiunea **Data**, adăugați un control de tip **BindingSource**. Acesta va avea numele implicit ***bindingSource1***.

Selectați controlul **bindingSouce1** din **try**-ul formei, iar în fereastra **Properties**, modificați proprietatea **DataSource** la **fermaDataSet1**, expandând nodurile ca în imaginea alăturată, apoi alegeți pentru proprietatea **DataMember** tabela **animale**.

Pentru a încărca în **numeListBox** toate numele animalelor existente în tabela **animale**, setaţi proprietatea **DataSource** a acestui control la ***bindingSouce1***, proprietatea **DisplayMember** la ***Nume***, proprietatea **ValueMember** la ***ID***.



Pentru a fi afişate în textbox-uri informaţiile despre animalul selectat în **numeListBox**, acestea trebuie să fie legate la **bindingSource1**, la fel ca numeListBox. Pentru aceasta, selectaţi pe rând textbox-urile şi în fereastra Properties, expandaţi nodul **DataBindings** aflat deasupra proprietăţii **Name** şi deschideţi lista corespunzătoare proprietăţii **Text** a acestor controale, expandaţi nodul ***bindingSouce1*** şi alegeţi câmpul din care se vor prelua datele afişate în textBox. *De exemplu:* pentru **nmTextBox** se va alege câmpul **Nume**, ca în imaginea alăturată. Faceţi setările pentru toate controalele de tip textBox aflate în partea dreaptă a formei.

Reveniţi în **Form1** şi pentru a trata evenimentul **Click** al opţiunii ***Dupa nume***, din meniul **Cautari**,veți scrie codul următor:

private void dupaNumeToolStripMenuItem\_Click(object sender, EventArgs e)

{

CautareDupaNume c = new CautareDupaNume();

c.ShowDialog();

}